Trajectory Planning in Dynamic Workspace: a 'State-Time Space' Approach

Thierry Fraichard

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Trajectory Planning in Dynamic Workspace: a 'State-Time Space' Approach

Thierry Fraichard

N° 3545

Octobre 1998

THÈME 3

ISSN 0249-6399

INRIA
RHÔNE-ALPES

Trajectory Planning in Dynamic Workspace: a 'State-Time Space' Approach

Thierry Fraichard*

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Sharp

Rapport de recherche n ° 3545 — Octobre 1998 — 25 pages

Abstract: This report addresses *trajectory planning in dynamic workspace*, *i.e.* motion planning for a robot subject to dynamic constraints and moving in a workspace with moving obstacles. First is introduced the novel concept of *state-time space*, *i.e.* the state space of the robot augmented of the time dimension. Like configuration space which is a tool to formulate path planning problems, state-time space is a tool to formulate trajectory planning in dynamic workspace problems. It permits to study the different aspects of dynamic trajectory planning, *i.e.* moving obstacles and dynamic constraints, in a unified way. Then this new concept is applied to the case of a car-like robot subject to dynamic constraints and moving along a given path on a dynamic planar workspace. A near-time-optimal approach that searches the solution trajectory over a restricted set of *canonical trajectories* is presented. These canonical trajectories are defined as having discrete and piecewise constant acceleration. Under these assumptions, it is possible to transform the problem of finding the time-optimal canonical trajectory to finding the shortest path in a directed graph embedded in the state-time space.

Key-words: trajectory planning, dynamic constraints, moving obstacles.

(Résumé : tsvp)

* thierry.fraichard@inria.fr

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN (France)
Téléphone : 04 76 61 52 00 - International: +33 4 76 61 52 00
Télécopie : 04 76 61 52 52 - International: +33 4 76 61 52 52

Planification de trajectoire en espace de travail dynamique: une approche basée sur "l'espace des états-temps"

Résumé : Nous abordons dans ce rapport la *planification de trajectoire en espace de travail dynamique*, *i.e.* la planification de mouvement pour un robot soumis à des contraintes dynamiques et qui se déplace dans un environnement comportant des obstacles mobiles. Nous commençons par introduire le concept "d'espace des états-temps", *i.e.* l'espace des états du robot augmenté de la dimension temporelle. De même que l'espace des configurations est un outil pour traiter la planification de chemin, l'espace des états-temps est un outil pour traiter la planification de trajectoire en espace de travail dynamique. Il permet d'aborder les différents aspects du problème, *i.e.* contraintes dynamiques et obstacles mobiles, dans un cadre unifié. Ce nouveau concept est alors mis en œuvre dans le cas d'un robot de type voiture se déplaçant le long d'un chemin donné dans un environnement planaire comportant des obstacles mobiles. Nous proposons une méthode qui recherche une trajectoire solution sous-optimale parmi un ensemble restreint de *trajectoires canoniques* définies comme ayant des accélérations discrètes et constantes par morceaux. La définition de ces trajectoires canoniques permet de transformer le problème de planification de trajectoire en un problème de recherche de plus court chemin dans un graphe défini dans l'espace des états-temps.

Mots-clé : planification de trajectoire, contraintes dynamiques, obstacles mobiles.

Contents

1 Introduction

1.1 Trajectory Planning in Dynamic Workspaces

A robot is designed to perform actions in its workspace (moving around, grasping and mating parts, etc.). Any such action usually implies that a motion is made by the robot. This accounts for the importance of motion planning in Robotics. This importance is naturally reflected in the number and the variety of research works dealing with motion planning (the reader is referred to [Latombe, 1990] for a survey of this topic). These works can be classified according to the type of motions that are planned. Thus it is possible to differentiate between *path planning* which is characterized by the search of a continuous sequence of configurations[1] between the current configuration of the robot and its goal configuration, and *trajectory planning* which is concerned with the time history of such a sequence.

Path planning is restricted to the geometric aspects of motion planning. The only constraints that can be taken into account are time-independent constraints such as stationary obstacles and kinematic constraints, *i.e.* constraints involving the configuration parameters of the robot and their derivatives. Depending on whether it is integrable, a kinematic constraint either reduces the set of allowed configurations (like an obstacle) or restricts the geometric shape of feasible paths.

On the other hand, trajectory planning with its time dimension permits to take into account time-dependent constraints such as moving obstacles and the dynamic constraints of the robot, *i.e.* the constraints imposed by the dynamics of the robot and the capabilities of its actuators.

Path planning has been extensively studied in the past twenty years. Whereas less attention has been paid to trajectory planning. And yet, when planning the motion of an actual robot, it is important to take into account the various constraints that restrict its motion capabilities and especially dynamic constraints. It is also important to deal with moving obstacles since an actual workspace will often be dynamic, *i.e.* with moving obstacles. Accordingly, this report addresses trajectory planning in dynamic workspaces, *i.e.* trajectory planning for a robot subject to dynamic constraints and moving in a dynamic workspace.

1.2 Related Works

A general approach that deals with moving obstacles is the configuration-time space approach which consists in adding the time dimension to the robot's configuration space [Erdmann and Lozano-Perez, 1987]. The robot maps in this configuration-time space to a point moving among stationary obstacles. Accordingly the different approaches developed in order to solve the path planning problem in the configuration space can be adapted in order to deal with the specificity of the time dimension and used (see [Latombe, 1990]). Among the existing works are those based upon extensions of the visibility graph [Erdmann and

[1] The *configuration* of a robot is a set of independent parametres that uniquely defines the position and orientation of every point of the robot.

Lozano-Perez, 1987; Fujimura and Samet, 1990; Reif and Sharir, 1985] and those based upon cell decomposition [Fujimura and Samet, 1989; Shih *et al.*, 1990].

There are several results for time-optimal trajectory planning for Cartesian robots subject to bounds on their velocity and acceleration [Canny *et al.*, 1990; Ó'Dúnlaing, 1987]. Besides optimal control theory provides some exact results in the case of robots with full dynamics and moving along a given path [Bobrow *et al.*, 1985b; Shiller and Dubowsky, 1985; Shiller and Lu, 1990]. Using these results, some authors have described methods that computes a local time-optimal trajectory [Shiller and Dubowsky, 1989; Shiller and Chen, 1990]. The key idea of these works is to formulate the problem as a two-stage optimization process: optimal motion time along a given path is used as a cost function for a local path optimization (hence local time-optimality). However the difficulty of the general problem and the need for practical algorithms led some authors to develop approximate methods. Their basic principle is to define a grid which is searched in order to find a near-time-optimal solution. Such grids are defined either in the workspace [Shiller and Dubowsky, 1988], the configuration space [Sahar and Hollerbach, 1985], or the state space of the robot [Canny *et al.*, 1988; Donald and Xavier, 1990; Jacobs *et al.*, 1989].

Few research works take into account moving obstacles and dynamic constraints simultaneously, and they usually do so with far too simplifying assumptions, *e.g.* [Fujimura and Samet, 1989] and [Ó'Dúnlaing, 1987]. More recently, [Fiorini and Shiller, 1996] has presented a two-stage algorithm that computes a local time-optimal trajectory for a manipulator arm with full dynamics and moving in a dynamic workspace: the solution is computed by first generating a collision-free path using the concept of velocity obstacle, and then by optimizing it thanks to dynamic optimization.

1.3 Contribution of the Report

The first contribution of this report is the concept of **state-time space** which is a tool to formulate problems of trajectory planning in dynamic workspaces. In this respect, it is similar to the concept of configuration space [Lozano-Perez and Wesley, 1979] which is a tool to formulate path planning problems. State-time space permits to study the different aspects of dynamic trajectory planning, *i.e.* moving obstacles and dynamic constraints, in a unified way. It stems from two concepts which have been used before in order to deal respectively with moving obstacles and dynamic constraints, namely the concepts of *configuration-time space* [Erdmann and Lozano-Perez, 1987], and *state space*, *i.e.* the space of the configuration parameters and their derivatives. Merging these two concepts leads naturally to state-time space, *i.e.* the state space augmented of the time dimension. In this framework, the constraints imposed by both the moving obstacles and the dynamic constraints can be represented by static forbidden regions of state-time space. Besides a trajectory maps to a curve in state-time space hence trajectory planning in dynamic workspaces simply consists in finding a curve in state-time space, *i.e.* a continuous sequence of state-times between the current state of the robot and a goal state. Such a curve must obviously respect additional constraints due to the fact that time is irreversible and that velocity and acceleration constraints translate to geometric constraints on the slope and the

curvature along the time dimension. However it is possible to extend previous methods for path planning in configuration space in order to solve the problem at hand.

The second contribution of this report is a method to solve trajectory planning in dynamic workspaces problems when cast in the state-time space framework. It is derived from a method originally presented in [Canny *et al.*, 1988], and extended to take into account the time dimension of the state-time space. It follows the paradigm of near-time-optimization: the search for the solution trajectory is performed over a restricted set of *canonical trajectories* hence the near-time-optimality of the solution. These canonical trajectories are defined as having piecewise constant acceleration that change its value at given times. Besides the acceleration is selected so as to be either minimum, null or maximum (bang controls). Under these assumptions, it is possible to transform the problem of finding the time-optimal canonical trajectory to finding the shortest path in a directed graph embedded in the state-time space.

1.4 Outline of the Report

§2.1, §2.2 and §2.3 describe the different features of the problem, *i.e.* the path $\mathcal{S}$, the robot $\mathcal{A}$ and the moving obstacles. Afterwards §2.4 gives a formal statement of the state-time space of $\mathcal{A}$ and of the problem which is to be solved. Finally §3 presents the algorithm developed in order to solve the problem at hand along with experimental results.

2 The Problem

The concept of state-time space was first introduced in [Fraichard and Laugier, 1992] to plan the motion of a point robot subject to simple velocity and acceleration bounds and moving along a given path amidst moving obstacles. Later, a mobile robot subject to full dynamic constraints was considered; first, in the case of a one-dimensional motion along a given path [Fraichard, 1993], and then, in the case of a two-dimensional motion on a planar surface [Fraichard and Scheuer, 1994].

In this report, it has been decided to focus on the case of a car-like robot with full dynamics and moving along a given path. The main reason for this choice is that, in this particular case, the state-time space is three-dimensional thus permitting a clear presentation of the concept of state-time space. Besides, although one-dimensional only, this motion planning problem does feature all the key characteristics of trajectory planning in dynamic workspaces, *i.e.* full dynamics and moving obstacles, and the concepts presented hereafter can easily be extended to problems of higher dimension (see for instance [Fraichard and Scheuer, 1994]). Accordingly the method presented here could readily be used within a path-velocity decomposition scheme [Kant and Zucker, 1986] to plan motions along a given path taking into account the robot dynamics (as in [Bobrow *et al.*, 1985a] or [Shin and McKay, 1985]) or moving obstacles (as in [Kyriakopoulos and Saridis, 1991] or [Ó'Dúnlaing, 1987]).

In summary, this report addresses trajectory planning for a car-like robot $\mathcal{A}$ which moves along a given path $\mathcal{S}$ on a planar workspace $\mathcal{W}$ cluttered up with stationary and moving obstacles. It is assumed that $\mathcal{S}$ is collision-free with the stationary obstacles of $\mathcal{W}$ and that it is feasible, *i.e.* that it respects the kinematic constraints that restricts the motion capabilities of $\mathcal{A}$. The problem then is to compute a trajectory for $\mathcal{A}$ that follows $\mathcal{S}$, is collision-free with the moving obstacles of $\mathcal{W}$ and satisfies the dynamic constraints of $\mathcal{A}$.

§2.1, §2.2 and §2.3 respectively describe the different features of the problem at hand, *i.e.* the path $\mathcal{S}$, the robot $\mathcal{A}$, and the moving obstacles. Afterwards §2.4 presents the state-time space of $\mathcal{A}$. Finally §2.5 gives a formal statement of the problem which is to be solved.

2.1 The Path $\mathcal{S}$

As mentioned earlier, the car-like robot $\mathcal{A}$ moves along a given path $\mathcal{S}$ which is collision-free with the stationary obstacles of $\mathcal{W}$ and respects the kinematic constraints of $\mathcal{A}$. In this section, we start by briefly presenting the kinematics of a car-like robot so as to derive the constraints that are taken into account in the definition of $\mathcal{S}$.

2.1.1 Kinematics of a Car-Like Robot

A car-like robot has two rear wheels and two directional front wheels. It is assumed that it moves on the plane $\mathbb{R}^2$. The configuration of a car is uniquely defined by the triple $(x, y, \theta) \in \mathbb{R}^2 \times [0, 2\pi[$ where (x, y) are the coordinates of the rear axle midpoint R and θ the orientation of the car (Fig. 1).

A body moving on the plane has only one centre of rotation. Let G be this centre of rotation. Assuming pure rolling condition, a wheel can only move in a direction which is normal to its axle. Therefore, when a car is moving, the axles of its wheels intersect at G. The orientation of the rear wheels being fixed, G is located on the rear wheels axle (possibly at an infinite distance) and R moves in a direction which is normal to this axle. In other words, the following constraint holds :

$$\tan\theta \quad = \quad \dot{y}/\dot{x} \tag{1}$$

Besides, due to the fact that the front wheels orientation is mechanically limited, the distance ρ between R and G, *i.e.* the curvature radius at point R, is lower bounded:

$$\rho \quad \geq \quad \rho_{\min} \tag{2}$$

Relations (1) and (2) are non-holonomic [Barraquand and Latombe, 1989]. As we will see further down, they restrict the geometric shape of the paths that are feasible for a car.

2.1.2 Defining $\mathcal{S}$

A path for a car is a continuous sequence of configurations, *i.e.* a curve in the $(xy\theta)$-space, that must verify the non-holonomic constraints (1) and (2). However, because of (1), a path

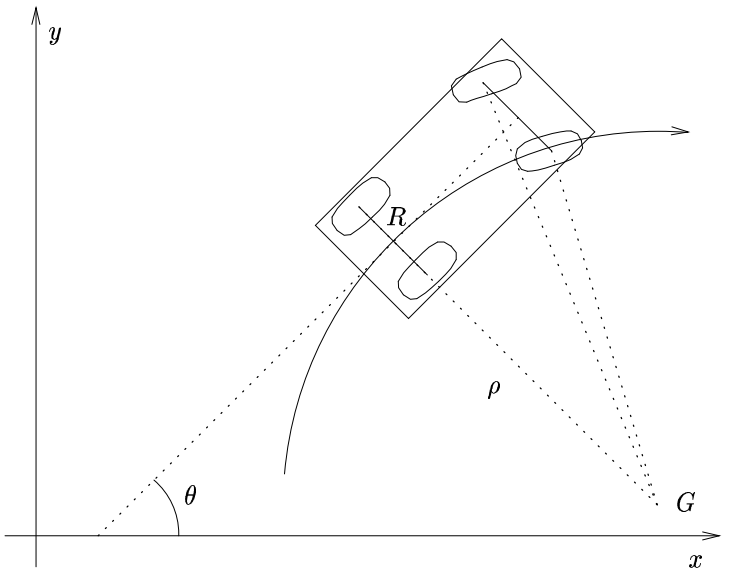

Figure 1: a car-like robot.

for a car is also defined by the xy-curve followed by R. Let Υ denote such a path, it also has to verify (1) and (2).

As a consequence of (1), Υ must be piecewise of class C^1 (a curve is of class C^n if it is differentiable n times and if its n^{th} derivative is continuous). Besides (2) implies that the curvature of Υ (wherever it is defined) must be upper-bounded by $1/\rho_{\min}$ and that the cusp points of Υ should correspond to inversion of the direction of motion of the car (back-up manœuvres). A path which respects these constraints is feasible but, it is important to note that the car has to stop at each cusp point (so as to change its direction of motion) and whenever a curvature discontinuity occurs (so as to change its front wheels' orientation).

Our main concern being in planning 'high' speed and forward motions only, $\mathcal{S}$ is defined as a planar curve of class C^2 whose curvature is upper-bounded by $1/\rho_{\min}$. The C^2 property insures that the path is manœuvre-free and that $\mathcal{A}$ can follow it without having to stop (no cusp points and no curvature discontinuity). The reader interested in the topic of C^2-path planning for car-like robots is referred to [Scheuer and Fraichard, 1997].

Assuming that $\mathcal{A}$ moves along $\mathcal{S}$, it is possible to reduce a configuration of $\mathcal{A}$ to the single variable s which represents the distance traveled along $\mathcal{S}$.

2.2 The Robot $\mathcal{A}$

In this section, we start by presenting the dynamic model of $\mathcal{A}$ that is used[2]. Then we describe the dynamic constraints that are taken into account.

2.2.1 The Dynamic Model of $\mathcal{A}$

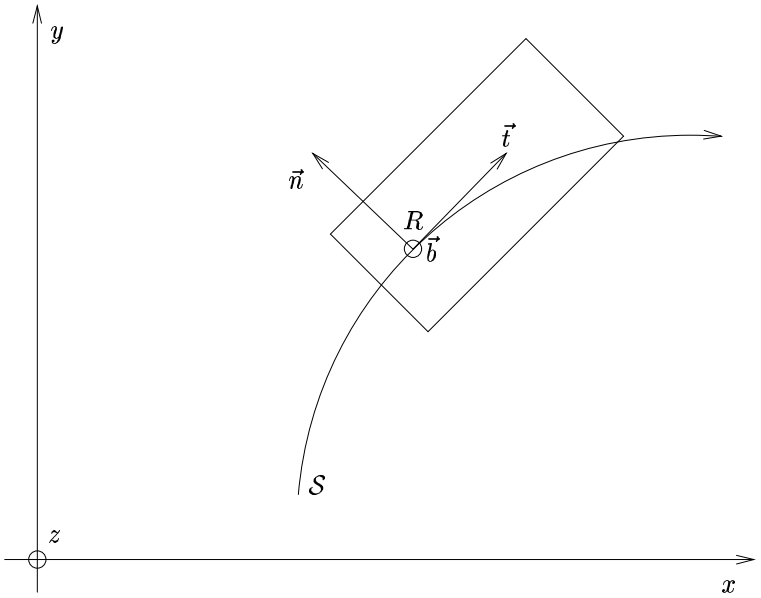

Figure 2: the frame attached to $\mathcal{A}$.

$\mathcal{A}$ is modelled as a rigid body supported by four wheels with rigid suspensions. Without loss of generality, it is assumed that the $\vec{t}$ axis of the frame attached to $\mathcal{A}$ coincides with the unit vector tangent to the path $\mathcal{S}$ at point R (Fig. 2). The $\vec{b}$ axis points in the positive direction normal to the plane. The $\vec{n}$ axis is chosen so that $(\vec{t}, \vec{n}, \vec{b})$ is right-handed. Note that the line of the radius of curvature at point R coincides with $\vec{n}$.

The motion of $\mathcal{A}$ along $\mathcal{S}$ obeys Newtonian dynamics. The external forces acting on $\mathcal{A}$ are the gravity force $\vec{G}$ and the ground reaction $\vec{R}$ which can be decomposed into their perpendicular components:

$$\vec{G} = -mg\,\vec{b} \tag{3}$$

$$\vec{R} = R_t\,\vec{t} + R_n\,\vec{n} + R_b\,\vec{b} \tag{4}$$

[2]This model is the two-dimensional instance of the model presented in [Shiller and Chen, 1990].

where m is the mass of $\mathcal{A}$ and g the gravity constant. The equation of motion of $\mathcal{A}$ can be expressed in terms of the tangential velocity $\dot{s}$ and the tangential acceleration $\ddot{s}$, namely:

$$\vec{G} + \vec{R} = m\ddot{s}\ \vec{t} + m\kappa_s \dot{s}^2\ \vec{n}$$

where κ_s is the signed curvature of the path at position s (κ_s is positive if the radial direction coincides with $\vec{n}$ and negative otherwise, $-1/\rho_{\min} \leq \kappa_s \leq 1/\rho_{\min}$). Using (3) and (4), this equation can be rewritten in the following set of equations:

$$R_t = m\ddot{s} \quad (5)$$

$$R_n = m\kappa_s \dot{s}^2 \quad (6)$$

$$R_b = mg \quad (7)$$

Equations (5) to (7) represent the forces required to maintain the velocity $\dot{s}$ and the acceleration $\ddot{s}$ of $\mathcal{A}$ at a given position s along the path. Although simple, this model is rich enough in the sense that the constraints associated are truly dynamic (they lead to state-dependence of the set of allowable accelerations).

2.2.2 The Dynamic Constraints of $\mathcal{A}$

Three dynamic constraints are taken into account (engine force, sliding and velocity constraints). They are presented in the next three sections. Afterwards they are transformed into constraints on the tangential velocity $\dot{s}$ and the tangential acceleration $\ddot{s}$.

Engine Force Constraint. When the robot is moving, the torque applied by the engine on the wheels translates into a planar force F whose direction is $\vec{t}$ and whose modulus is $m\ddot{s}$. This force is bounded by the maximum (resp. minimum) equivalent engine force:

$$F_{\min} \leq F \leq F_{\max} \quad (8)$$

These bounds are assumed to be constant and independent of the speed.

Sliding Constraint. The component of $\vec{R}$ in the plane $\vec{t} \times \vec{n}$ represents the friction that is applied from the ground to the wheels. This friction is constrained by the following relation:

$$\sqrt{{R_t}^2 + {R_n}^2} \leq \mu R_b \quad (9)$$

where μ is the friction coefficient between the wheels and the ground. If this constraint is violated then $\mathcal{A}$ will slide off the path.

Velocity Constraint. Our main constraint being in planning forward motions, the velocity $\dot{s}$ is constrained by the following relation:

$$0 \leq \dot{s} \leq \dot{s}_{\max} \quad (10)$$

where $\dot{s}_{\max}$ is the highest velocity allowed.

Tangential Acceleration Constraints. The engine force constraint (8) yields the following feasible acceleration range:

$$\frac{F_{\min}}{m} \leq \ddot{s} \leq \frac{F_{\max}}{m} \tag{11}$$

Besides substituting (5), (6) and (7) in (9) and solving it for $\ddot{s}$ yields the following relation which expresses the feasible acceleration range due to the sliding constraint:

$$-\sqrt{\mu^2 g^2 - {\kappa_s}^2 \dot{s}^4} \leq \ddot{s} \leq \sqrt{\mu^2 g^2 - {\kappa_s}^2 \dot{s}^4} \tag{12}$$

The final feasible acceleration range is therefore given by the intersection of (11) and (12):

$$\max\left(\frac{F_{\min}}{m}, -\sqrt{\mu^2 g^2 - {\kappa_s}^2 \dot{s}^4}\right) \leq \ddot{s} \leq \min\left(\frac{F_{\max}}{m}, \sqrt{\mu^2 g^2 - {\kappa_s}^2 \dot{s}^4}\right) \tag{13}$$

Tangential Velocity Constraints. Velocity $\dot{s}$ must respect (10). Besides the argument under the square roots in (12) should be positive. When $\kappa_s \neq 0$, $\dot{s}$ must respect the following constraint:

$$-\sqrt{\frac{\mu g}{|\kappa_s|}} \leq \dot{s} \leq \sqrt{\frac{\mu g}{|\kappa_s|}} \tag{14}$$

The final feasible velocity range is therefore given by the intersection of (10) and (14):

$$0 \leq \dot{s} \leq \min\left(\dot{s}_{\max}, \sqrt{\frac{\mu g}{|\kappa_s|}}\right) \tag{15}$$

The latter constraint can be expressed as a set of forbidden states, *i.e.* points of the $s \times \dot{s}$ plane. Let $\mathcal{TV}$ be this set of states, it is defined as:

$$\mathcal{TV} = \left\{(s, \dot{s}) \mid 0 > \dot{s} \text{ or } \dot{s} > \min\left(\dot{s}_{\max}, \sqrt{\frac{\mu g}{|\kappa_s|}}\right)\right\}$$

2.3 The Moving Obstacles

$\mathcal{A}$ moves in a workspace $\mathcal{W} \in \mathbb{R}^2$ which is cluttered up with stationary and moving obstacles. The path $\mathcal{S}$ being collision-free with the stationary obstacles, only the moving obstacles have to be considered when it comes to planning $\mathcal{A}$'s trajectory.

Let $\mathcal{B}_i, i \in \{1, \ldots, m\}$, be the set of moving obstacles. Let $\mathcal{B}_i(t)$ denotes the region of $\mathcal{W}$ occupied by $\mathcal{B}_i$ at time t and $\mathcal{A}(s)$ the region of $\mathcal{W}$ occupied by $\mathcal{A}$ at position s along $\mathcal{S}$. If, at time t, $\mathcal{A}$ is at position s and if there is an obstacle $\mathcal{B}_i$ such that $\mathcal{B}_i(t)$ intersects $\mathcal{A}(s)$ then a collision occurs between $\mathcal{A}$ and $\mathcal{B}_i$. Accordingly the constraints imposed by the moving obstacles on $\mathcal{A}$'s motion can be represented by a set of forbidden points of the $s \times t$ plane. Let $\mathcal{TB}$ be this set of forbidden points, it is defined as:

$$\mathcal{TB} = \{(s, t) \mid \exists i \in \{1, \ldots, m\},\ \mathcal{A}(s) \cap \mathcal{B}_i(t) \neq \emptyset\}$$

2.4 The State-Time Space of $\mathcal{A}$

As mentioned earlier, the configuration of $\mathcal{A}$ is reduced to the single variable s which represents the distance traveled along $\mathcal{S}$. A state of $\mathcal{A}$ is therefore represented by a pair $(s, \dot{s}) \in [0, s_{\max}] \times [0, \dot{s}_{\max}]$ where $s_{\max}$ is the arc-length of $\mathcal{S}$.

A **state-time** of $\mathcal{A}$ is defined by adding the time dimension to a state hence it is represented by a triple $(s, \dot{s}, t) \in [0, s_{\max}] \times [0, \dot{s}_{\max}] \times [0, \infty)$. The set of every state-time is the **state-time space** of $\mathcal{A}$, it is denoted by $\mathcal{ST}$.

A state-time is admissible if it does not violate the no-collision and velocity constraints presented earlier. Before defining an admissible state-time formally, let us define $\mathcal{TB}'$, the set of state-times which entail a collision between $\mathcal{A}$ and a moving obstacle. $\mathcal{TB}'$ is simply derived from $\mathcal{TB}$:

$$\mathcal{TB}' = \{(s, \dot{s}, t) \mid \exists i \in \{1, \ldots, m\},\ \mathcal{A}(s) \cap \mathcal{B}_i(t) \neq \emptyset\}$$

Similarly we define $\mathcal{TV}'$, the set of state-times which violate the velocity constraint (15). $\mathcal{TV}'$ is simply derived from $\mathcal{TV}$:

$$\mathcal{TV}' = \left\{ (s, \dot{s}, t) \mid 0 > \dot{s} \text{ or } \dot{s} > \min\left(\dot{s}_{\max}, \sqrt{\frac{\mu g}{|\kappa_s|}}\right) \right\}$$

Accordingly a state-time q is **admissible** if and only if:

$$q \in \mathcal{ST} \setminus (\mathcal{TB}' \cup \mathcal{TV}')$$

where $E \setminus F$ denotes the complement of F in E. The set of every admissible state-time is the **admissible state-time space** of $\mathcal{A}$, it is denoted by $\mathcal{AST}$ and defined as:

$$\mathcal{AST} = \mathcal{ST} \setminus (\mathcal{TB}' \cup \mathcal{TV}')$$

Figure 3 depicts the state-time space of $\mathcal{A}$ in a simple case where there is only one moving obstacle which crosses $\mathcal{S}$.

In this framework, a *trajectory* Γ for $\mathcal{A}$ between an initial state $(s_i,\ \dot{s}_i)$ and a final state $(s_f,\ \dot{s}_f)$ can be represented by a curve of $\mathcal{ST}$, *i.e.* a continuous sequence of state-times between the initial state-time $(s_i,\ \dot{s}_i,\ 0)$ and a final state-time $(s_f,\ \dot{s}_f,\ t_f)$. t_f is the duration of the trajectory Γ. The acceleration profile of Γ is a continuous map $\ddot{s} : [0, t_f] \longrightarrow \mathbb{R}$. $\ddot{s}(t)$ represents the acceleration which is applied to $\mathcal{A}$ at time t. Note that the velocity $\dot{s}$ and position s of $\mathcal{A}$ along $\mathcal{S}$ are respectively defined as the first and second integral of $\ddot{s}$ subject to an initial position and velocity. In order to be feasible, Γ has to verify the different constraints presented in the previous sections, *i.e.* it must be collision-free with the moving obstacles and respect (13) and (15). Figure 4 depicts an example of trajectory between $(s_i,\ \dot{s}_i)$ and $(s_f,\ \dot{s}_f)$.

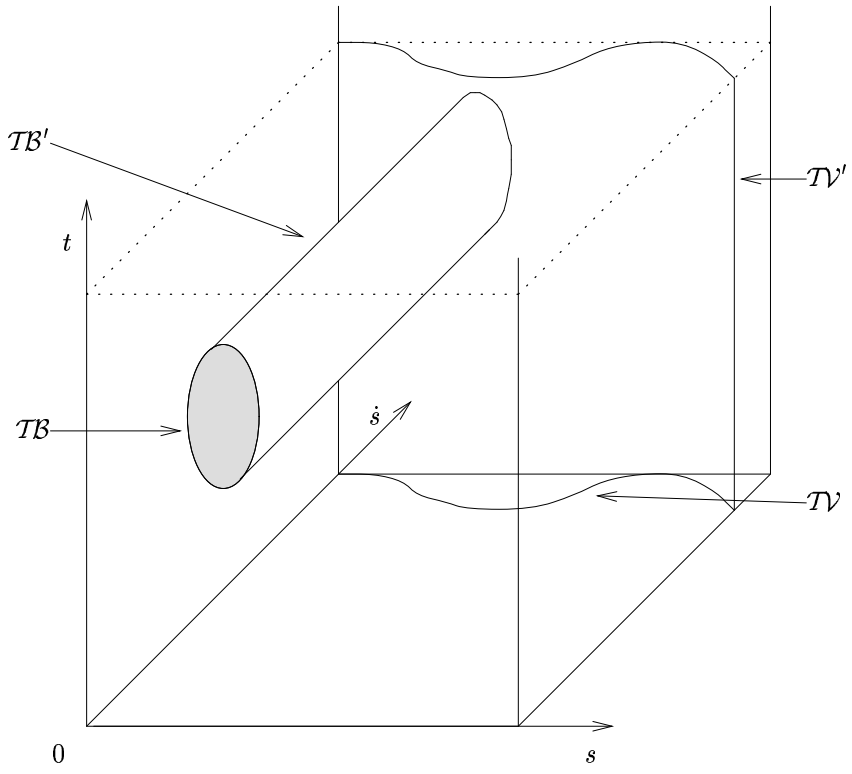

Figure 3: $\mathcal{ST}$, the state-time space of $\mathcal{A}$.

2.5 Statement of the Problem

Finally, we can formally state the problem which is to be solved. Let $(s_i,\ \dot{s}_i)$ be the start state of $\mathcal{A}$ and $(s_f,\ \dot{s}_f)$ its goal state. A trajectory $\Gamma : [0,1] \longrightarrow \mathcal{ST}$ is a solution to the problem at hand if and only if:

1. $\Gamma(0) = (s_i, \dot{s}_i, 0)$ and $\Gamma(1) = (s_f, \dot{s}_f, t_f)$.
2. $\Gamma \subset \mathcal{AST}$.
3. Γ's acceleration profile respects (13).

Naturally, we are interested in finding a time-optimal trajectory, *i.e.* a trajectory such that t_f should be minimal.

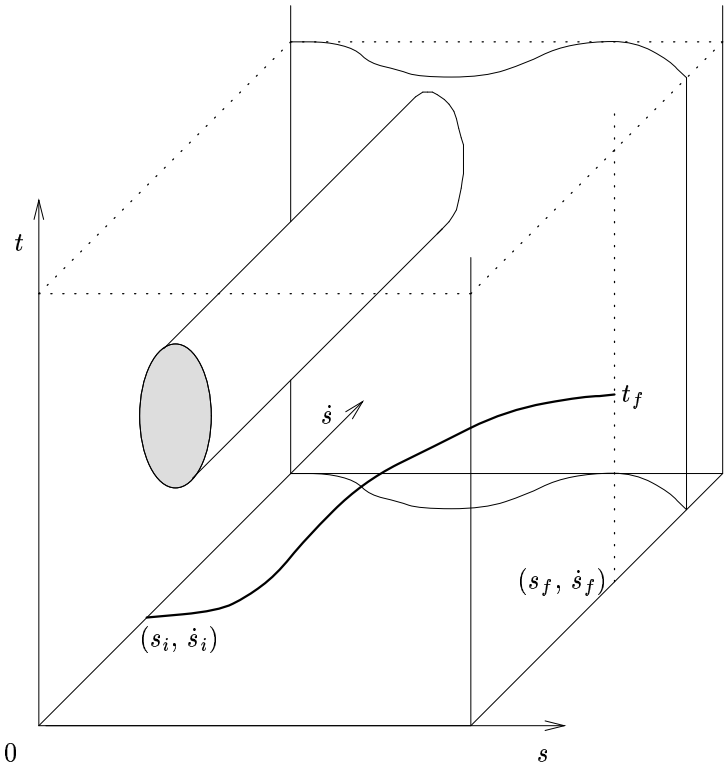

Figure 4: a trajectory between $(s_i, \dot{s}_i)$ and $(s_f, \dot{s}_f)$.

3 A Solution Algorithm

3.1 Outline of the Approach

The method that we have developed in order to solve the problem at hand, *i.e.* to find a curve Γ of the state-time space $\mathcal{ST}$ which respect the various constraints presented in the previous section, was initially motivated by the work described in [Canny *et al.*, 1988]. For reasons which will be discussed later in §3.6, we follow the paradigm of near-time-optimization, *i.e.* instead of trying to find out the exact time-optimal trajectory between an initial and a final state, we compute an approximate time-optimal solution by performing the search over a restricted set of *canonical trajectories*. These canonical trajectories are defined as having piecewise constant acceleration $\ddot{s}$ that can only change its value at given times $k\tau$ where τ is a time-step and k some positive integer. Besides $\dot{s}$ is selected so as to

be either minimum, null or maximum. Under these assumptions, it is possible to transform the problem of finding the time-optimal canonical trajectory to finding the shortest path in a directed graph $\mathcal{G}$ embedded in $\mathcal{ST}$. The vertices $\mathcal{G}$ form a regular grid embedded in $\mathcal{ST}$ while the edges corresponds to canonical trajectory segments that each takes time τ. The next sections respectively present the canonical trajectories, the graph $\mathcal{G}$, the search algorithm and experimental results. Finally we discuss the interest of such an approach.

3.2 The Canonical Trajectories

The definition of the canonical trajectories depends on discretizing time — a time-step τ is chosen — and selecting an acceleration $\ddot{s}$ that respects the acceleration constraint (13) and which is either minimum, null or maximum. From a practical point of view, the set of accelerations is discretized — an acceleration-step δ is chosen — and the acceleration applied to $\mathcal{A}$ at each time-step, *i.e.* the minimum, null or maximum one, is selected from this discrete set. As we will see further down, this discretization yields a regular grid in $\mathcal{ST}$.

First let us determine the minimum (resp. maximum) acceleration $\ddot{s}_{\min}$ (resp. $\ddot{s}_{\max}$) that can be applied to $\mathcal{A}$. $\ddot{s}_{\min}$ and $\ddot{s}_{\max}$ are derived from (13) by noting that the acceleration that can be applied to $\mathcal{A}$ is maximum (resp. minimum) when the curvature κ_s is null, in other words:

$$\begin{aligned} \ddot{s}_{\min} &= \max\left(\frac{F_{\min}}{m}, -\sqrt{\mu^2 g^2}\right) \\ \ddot{s}_{\max} &= \min\left(\frac{F_{\max}}{m}, \sqrt{\mu^2 g^2}\right) \end{aligned}$$

The interval $[\ddot{s}_{\min}, \ddot{s}_{\max}]$ is therefore the overall range of accelerations that can be applied to $\mathcal{A}$. Now, when $\mathcal{A}$ follows a given path $\mathcal{S}$, at each time instant, it can withstand a range of accelerations that is a subset of $[\ddot{s}_{\min}, \ddot{s}_{\max}]$, this subset is derived from (13) and depends on the current curvature of $\mathcal{S}$. Given the acceleration step δ, Δ, the overall discrete set of accelerations that can be applied to $\mathcal{A}$ is defined as:

$$\Delta = \left\{ i\delta | i \in \mathbb{N}, \lceil \frac{\ddot{s}_{\min}}{\delta} \rceil \leq i \leq \lfloor \frac{\ddot{s}_{\max}}{\delta} \rfloor \right\}$$

Let $\Gamma : [0,1] \longrightarrow \mathcal{ST}$ be a trajectory and $\ddot{s} : [0, t_f] \longrightarrow \Delta$ its acceleration profile. Γ is a **canonical trajectory** if and only if:

- $\ddot{s}$ only changes its value at times $k\tau$ where $k \in \mathbb{N}, 0 \leq k \leq \lfloor t_f/\tau \rfloor$.
- Let $\ddot{s}^{k\tau}_{\min}$ (resp. $\ddot{s}^{k\tau}_{\max}$) be the minimum (resp. maximum) acceleration allowed w.r.t. the state of $\mathcal{A}$ at time $k\tau$. $\ddot{s}(k\tau)$ is chosen from Δ so as to be either null or as close as possible to $\ddot{s}^{k\tau}_{\min}$ and $\ddot{s}^{k\tau}_{\max}$. Thus we have:

$$\ddot{s}(k\tau) \in \{\delta\lceil \frac{\ddot{s}^{k\tau}_{\min}}{\delta} \rceil,\ 0,\ \delta\lfloor \frac{\ddot{s}^{k\tau}_{\max}}{\delta} \rfloor\}$$

As we will see further down in §3.4.3, $\ddot{s}^{k\tau}_{\min}$ and $\ddot{s}^{k\tau}_{\max}$ are computed so as to ensure that the acceleration constraint (13) is respected along the trajectory until the next acceleration change. Note that such a trajectory is very similar to the so-called 'bang-bang' trajectory of the control literature except that, in our case, the acceleration switches occur at regular time intervals.

3.3 The State-Time Graph $\mathcal{G}$

Let q be a state-time, *i.e.* a point of $\mathcal{ST}$. It is a triple $(s, \dot{s}, t)$. It can equivalently be represented by $q(t) = (s(t), \dot{s}(t))$. Let $q(k\tau) = (s(k\tau), \dot{s}(k\tau))$ be a state-time of $\mathcal{A}$ and $q((k+1)\tau)$ one of the state-times that $\mathcal{A}$ can reach by a canonical trajectory of duration τ. $q((k+1)\tau)$ is obtained by applying an acceleration $\ddot{s} \in \Delta$ to $\mathcal{A}$ for the duration τ. Accordingly we have:

$$\dot{s}((k+1)\tau) = \dot{s}(k\tau) + \ddot{s}\tau \tag{16}$$

$$s((k+1)\tau) = s(k\tau) + \dot{s}(k\tau)\tau + \frac{1}{2}\ddot{s}\tau^2 \tag{17}$$

By analogy with [Canny *et al.*, 1988], the trajectory between $q(k\tau)$ and $q((k+1)\tau)$ is called a $(\ddot{s}, \tau)$**-bang**. The state-time $q((k+1)\tau)$ is reachable from $q(k\tau)$. Obviously a canonical trajectory is made up of a sequence of $(\ddot{s}, \tau)$-bangs.

Let $q(m\tau)$, $m \geq k$, be a state-time reachable from $q(k\tau)$. Assuming that $\dot{s}(k\tau)$ is a multiple of $\delta\tau$, it can be shown that the following relations hold for some integers α_1 and α_2:

$$s(m\tau) = s(k\tau) + \alpha_1 \frac{1}{2}\delta\tau^2$$

$$\dot{s}(m\tau) = \dot{s}(k\tau) + \alpha_2\delta\tau$$

Thus all state-times reachable from one given state-time by a canonical trajectory lie on a regular grid embedded in $\mathcal{ST}$. This grid has spacings of $\delta\tau^2/2$ in position, of $\delta\tau$ in velocity and of τ in time.

Consequently it becomes possible to define a directed graph $\mathcal{G}$ embedded in $\mathcal{ST}$. The nodes of $\mathcal{G}$ are the grid-points while the edges of $\mathcal{G}$ are $(\ddot{s}, \tau)$-bangs between pairs of nodes. $\mathcal{G}$ is called the **state-time graph**, Let η be a node in $\mathcal{G}$, the state-times reachable from η by a $(\ddot{s}, \tau)$-bang lie on the grid, they are nodes of $\mathcal{G}$ (Fig. 5). An edge between η and one of its neighbours represents the corresponding $(\ddot{s}, \tau)$-bang. A sequence of edges between two nodes defines a canonical trajectory. The time of such a canonical trajectory is trivially equal to τ times the number of edges in the trajectory. Therefore the shortest path between two nodes (in term of number of edges) is the *time-optimal canonical trajectory* between these nodes.

Let $\mathbf{s} = (s_i, \dot{s}_i)$ be the initial state of $\mathcal{A}$ and $\mathbf{g} = (s_f, \dot{s}_f)$ be its goal state. Without loss of generality it is assumed that the corresponding initial state-time $\mathbf{s}^* = (s_i, \dot{s}_i, 0)$ and the corresponding set of goal state-times $\mathbf{G}^* = \{(s_f, \dot{s}_f, k\tau) \text{ with } k \geq 0\}$ are grid-points.

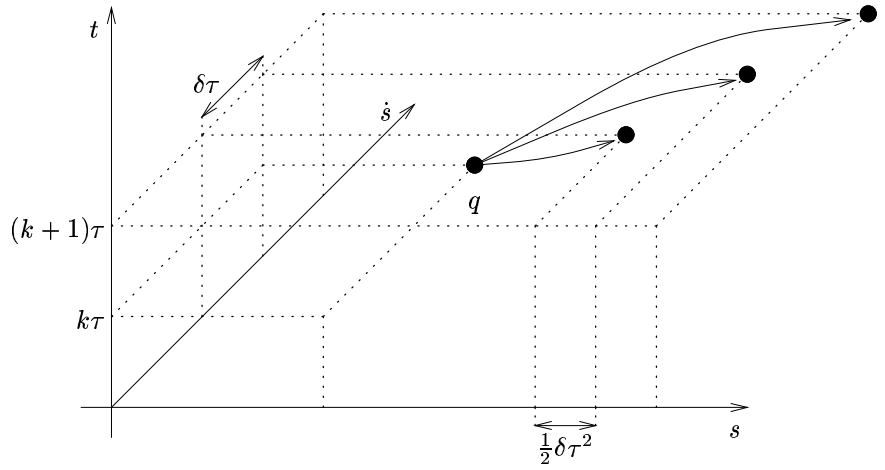

Figure 5: $\mathcal{G}$, the graph embedded in $\mathcal{ST}$.

Accordingly searching for a time-optimal canonical trajectory between **s** and **g** is equivalent to searching a shortest path in $\mathcal{G}$ between the node $\mathbf{s}^*$ and a node in $\mathbf{G}^*$.

From a practical point of view, the state-time graph $\mathcal{G}$ is embedded in a compact region of $\mathcal{ST}$. More precisely, the time component of the grid-points is upper bounded by a certain value $t_{\max}$ which can be viewed as a time-out. The number of grid-points is therefore finite and so is $\mathcal{G}$. Accordingly the search for the time-optimal canonical trajectory can be done in a finite amount of time.

3.4 Searching the State-Time Graph

3.4.1 The Algorithm

We use an A^* algorithm to search $\mathcal{G}$ [Nilsson, 1980]. Starting with $\mathbf{s}^*$ as the current node, we expand this current node, *i.e.* we determine all its neighbours, then we select the neighbour which is the best according to a given criterion (a cost function) and it becomes the current node. This process is repeated until the goal is reached or until the whole graph has been explored. The time-optimal path is returned using back-pointers. In the next two sections, we detail two key-points of the algorithm, namely the cost function assigned to each node and the node expansion.

3.4.2 The Cost Function

A^* assigns a cost $f(\eta)$ to every node η in $\mathcal{G}$. Since we are looking for a time-optimal path, we have chosen $f(\eta)$ as being the estimate of the time-optimal path in $\mathcal{G}$ connecting $\mathbf{s}^*$ to

$\mathbf{G}^*$ and passing through η. $f(\eta)$ is classically defined as the sum of two components $g(\eta)$ and $h(\eta)$:

- $g(\eta)$ is the duration of the path between $\mathbf{s}^*$ and η, *i.e.* the time component of η.
- $h(\eta)$ is the estimate of the time-optimal path between η and an element of $\mathbf{G}^*$, *i.e.* the amount of time it would take $\mathcal{A}$ to reach $\mathbf{g}$ from its current state with a 'bang-coast-bang' acceleration profile, *i.e.* maximum overall acceleration $\ddot{s}_{\max}$, null acceleration and minimum overall acceleration $\ddot{s}_{\min}$. When such an acceleration profile does not exist[3], $h(\eta)$ is set to $+\infty$.

The heuristic function $h(\eta)$ is trivially admissible, thus A^* is guaranteed to generate the time-optimal path whenever it exists [Nilsson, 1980]. Besides the fact that $f(\eta)$ is locally consistent improves the efficiency of the algorithm.

3.4.3 The Node Expansion

The neighbours of a given node $\eta = (s,\ \dot{s},\ \mathrm{k}\tau)$ are the nodes which can be reached from η by a $(\ddot{s},\tau)$-bang. As mentioned earlier, $\ddot{s} \in \{\lfloor \ddot{s}^{k\tau}_{\min} + \delta \rfloor,\ 0,\ \lceil \ddot{s}^{k\tau}_{\max} - \delta \rceil\}$. $\ddot{s}^{k\tau}_{\min}$ and $\ddot{s}^{k\tau}_{\max}$ have to be computed so as to ensure that the acceleration constraint (13) is respected along the corresponding $(\ddot{s},\tau)$-bang. This computation is done in a conservative way. First the farthest position, say s^+, that $\mathcal{A}$ can reach from its current state is determined. It is the position reached after a $(\ddot{s}_{\max},\tau)$-bang. Then the maximum curvature between s and s^+ is determined and substituted into (13) so as to yield the desired acceleration bounds $\ddot{s}^{k\tau}_{\min}$ and $\ddot{s}^{k\tau}_{\max}$. Finally it remains to check that the $(\ddot{s},\tau)$-bang associated with each of the candidate neighbours does not violate the velocity and collision avoidance constraints, *i.e.* that the $(\ddot{s},\tau)$-bang is included in $\mathcal{AST}$.

As we will see now, it is not necessary to compute $\mathcal{AST}$ to check these two points. Velocity checking is done by using the motion equation of $\mathcal{A}$, while collision checking is performed directly in $\mathcal{W}$. Let us consider a $(\ddot{s},\tau)$-bang taking place between time instants $k\tau$ and $(k+1)\tau$. $\forall t \in [0,\tau]$ and according to (16) and (17), we have:

$$\dot{s}(k\tau + t) = \dot{s}(k\tau) + \ddot{s}t \tag{18}$$

$$s(k\tau + t) = s(k\tau) + \dot{s}(k\tau)t + \frac{1}{2}\ddot{s}t^2 \tag{19}$$

Velocity Constraint. Using (18), it is straightforward to check that a $(\ddot{s},\tau)$-bang does not violate the velocity bounds (14) stated in §2.2.2.

Collision Avoidance. Recall that a $(\ddot{s},\tau)$-bang between $k\tau$ and $(k+1)\tau$ is collision-free if and only if:

$$\forall t \in [k\tau,(k+1)\tau], \forall i \in \{1,\ldots,m\},\ \mathcal{A}(s(t)) \cap \mathcal{B}_i(t) = \emptyset$$

[3] In this case, η is no longer reachable.

Eq. (19) provides the position of $\mathcal{A}$ at every time along the $(\ddot{s}, \tau)$-bang, and collision checking can efficiently be performed by computing the intersection between the two planar regions $\mathcal{A}(s(t))$ and $\mathcal{B}_i(t)$.

It might be desirable to add a safety margin to the collision checking procedure so as to incorporate the uncertainty on the motions of $\mathcal{A}$ and of the moving obstacles. In this case, the collision avoidance condition becomes:

$$\forall t \in [k\tau, (k+1)\tau], \forall i \in \{1, \ldots, m\},\ \mathcal{G}(\mathcal{A}(s(t)), sm) \cap \mathcal{B}_i(t) = \emptyset$$

where $\mathcal{G}(X, sm)$ denotes the planar region X isotropically grown of the safety margin sm. The safety margin can integrate both a fixed and a velocity-dependent term, *e.g.* $sm = c_0 + c_1 \dot{s}(t)$ with c_0 and $c_1 \in \mathbb{R}$.

3.4.4 Complexity Issues

Expanding a node of the graph $\mathcal{G}$ can be done efficiently in constant time (recall that the admissible state-time space $\mathcal{AST}$ is not computed, collision checking is performed directly in the two-dimensional workspace $\mathcal{W}$). The heuristic function used for the A^* search is both admissible and locally consistent (*cf.* §3.4.2), the time complexity of the A^* algorithm is therefore $O(n)$ where n is the number of vertices in $\mathcal{G}$ [Farreny and Ghallab, 1987]. n is defined as:

$$n = \frac{2 s_{\max}}{\delta \tau^2} \frac{\dot{s}_{\max}}{\delta \tau} \frac{t_{\max}}{\tau}$$

The acceleration step and, to a greater extent, the time step are key factors as far as the running time of the algorithm is concerned. Experimental running times are given in §3.5. A discussion about the choice of the discretization steps is offered in §3.6

3.5 Implementation and Experiments

The algorithm presented earlier has been implemented in C on a Sparc Station. Two examples of trajectory planning are depicted in Fig. 6 and 7. In each case, there are two windows: a trace window showing the part of the graph which has been explored and a result window displaying the final trajectory. Any such window represents the $s \times t$ plane (the position axis is horizontal while the time axis is vertical; the frame origin is at the upper-left corner). The thick black segments represent the trails left by the moving obstacles and the little dots are points of the underlying grid. Note that the vertical spacing of the dots corresponds to the time-step τ. In both examples, $\mathcal{A}$ starts from position 0 (upper-left corner) with a null velocity, it is to reach position $s_{\max}$ (right border) with a null velocity.

The values of the different parametres and discretization steps in these experiments are selected in order to simulate a car-like vehicle moving in the road network: $\dot{s}_{\max} = 20m/s, -\ddot{s}_{\min} = \ddot{s}_{\max} = 1m/s^2$. The idea is to plan the motion of the vehicle for the next 500m ($s_{\max} = 500m$). The time horizon $t_{\max}$ is set to 25s and the obstacles are assumed to keep a constant velocity over the time horizon. For a value of τ set of $0.5s$, the running time is of the order of $1s$.

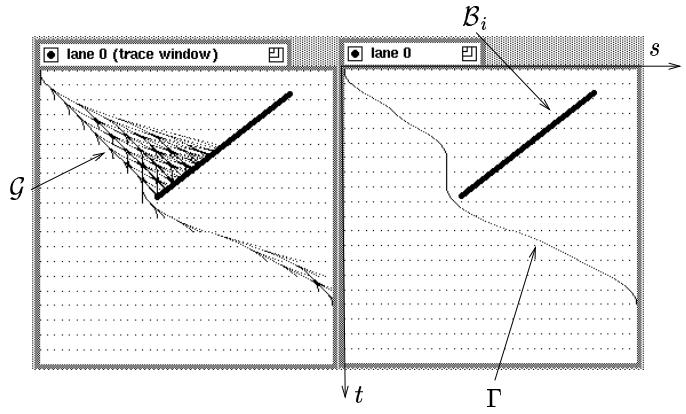

Figure 6: experimental results.

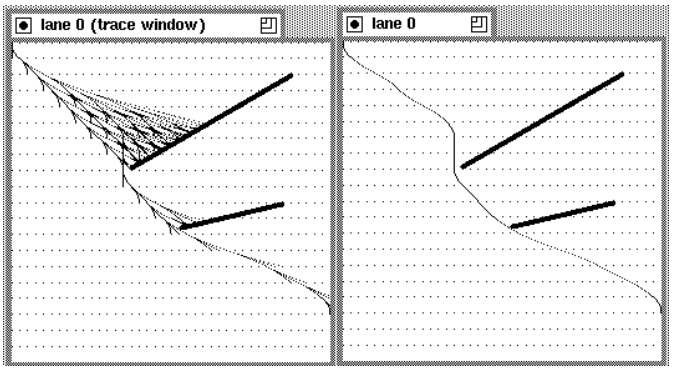

Figure 7: experimental results.

3.6 Discussion on the Proposed Solution

As mentioned above in §3.4.4, the running time of the search algorithm depends on the size of the graph $\mathcal{G}$ which is to be explored (number of nodes). In turn this size is directly related to the value of the time-step τ — the smaller τ, the higher the number of vertices in $\mathcal{G}$. On the other hand, we intuitively[4] feel that the quality of the solution trajectory is also related to the value of τ — the smaller τ, the better the approximation. Thus it is possible to trade off the computation speed against the quality of the solution.

This property is very important and we would like to advocate this type of approach when dealing with an actual dynamic workspace. In such a workspace, it is usually impossible to

[4]This intuition is confirmed in [Canny *et al.*, 1988] where it is shown that, for a correct choice of τ, any safe trajectory can be approximated to a tolerance ϵ by a safe canonical trajectory.

have a full a priori knowledge of the motion of the moving obstacles. It is more likely that the knowledge that we have of their motions be restricted to a certain time interval, *i.e.* a time horizon. This time horizon may represent the duration over which an estimation of the motions of the moving obstacles is sound. The main consequence of this assumption is to set an upper bound on the time available to plan the motion of our robot (in a highly dynamic workspace, this upper bound may be very low). In this case, an approach such as the one we have presented is most interesting because its average running time can be tuned w.r.t. the time horizon considered.

4 Conclusion

This report addressed *trajectory planning in dynamic workspaces* which is defined as trajectory planning for a robot subject to dynamic constraints and moving in a dynamic workspace, *i.e.* with moving obstacles.

First was introduced the novel concept of *state-time space*, *i.e.* the state space of the robot augmented of the time dimension. Like the concept of configuration space [Lozano-Perez and Wesley, 1979] which is a tool to formulate path planning problems, state-time space is a tool to formulate trajectory planning in dynamic workspaces problems. It permits to study the different aspects of dynamic trajectory planning, *i.e.* moving obstacles and dynamic constraints, in a unified way. Thus the constraints imposed by both the moving obstacles and the dynamic constraints can be represented by static forbidden regions of state-time space. Besides a trajectory maps to a curve in state-time space hence dynamic trajectory planning simply consists in finding a curve in state-time space, *i.e.* a continuous sequence of state-times between the current state of the robot and a goal state.

Then this new concept was applied to the case of a car-like robot subject to dynamic constraints and moving along a given path on a dynamic planar workspace. A near-time-optimal approach that searches the solution trajectory over a restricted set of *canonical trajectories* was presented. These canonical trajectories are defined as having piecewise constant acceleration that can only change its value at given times. Besides the acceleration is selected so as to be either minimum, null or maximum. Under these assumptions, it is possible to transform the problem of finding the time-optimal canonical trajectory to finding the shortest path in a directed graph embedded in the state-time space.

We believe that state-time space is a useful tool to address trajectory planning in dynamic workspaces problems. In this report, it was applied to the simple, yet rich enough, case of a car-like robot moving along a given path. However it has also been applied to the more complex case of a car-like robot moving on a planar surface [Fraichard and Scheuer, 1994]. Further work should permit to assess its usefulness.

Acknowledgements

This research work was partially supported by the French Ministry of Research and the European Eureka EU-153 Prometheus programme.

References

[Barraquand and Latombe, 1989] J. Barraquand and J-C Latombe. – On non-holonomic mobile robots and optimal maneuvering. – *Revue d'intelligence Artificielle*, 3(2):77–103, 1989.

[Bobrow *et al.*, 1985a] J. E. Bobrow, S. Dubowsky, and J. S. Gibson. – Time-optimal control of robotic manipulators along specified paths. – *Int. Journal of Robotics Research*, 4(3):3–17, Fall 1985.

[Bobrow *et al.*, 1985b] J.E. Bobrow, S. Dubowsky, and J.S. Gibson. – Time-optimal control of robotic manipulators along specified paths. – *Int. Journal of Robotics Research*, 4(3):3–17, Fall 1985.

[Canny *et al.*, 1988] J. Canny, B. Donald, J. Reif, and P. Xavier. – On the complexity of kynodynamic planning. – In *Proc. of the IEEE Symp. on the Foundations of Computer Science*, pages 306–316, White Plains, NY (USA), November 1988.

[Canny *et al.*, 1990] J. Canny, A. Rege, and J. Reif. – An exact algorithm for kinodynamic planning in the plane. – In *Proc. of the ACM Symp. on Computational Geometry*, pages 271–280, Berkeley, CA (USA), 1990.

[Donald and Xavier, 1990] B. Donald and P. Xavier. – Provably good approximation algorithms for optimal kinodynamic planning for cartesian robots and open-chain manipulators. – In *Proc. of the ACM Symp. on Computational Geometry*, pages 290–300, Berkeley, CA (USA), 1990.

[Erdmann and Lozano-Perez, 1987] M. Erdmann and T. Lozano-Perez. – On multiple moving objects. – *Algorithmica*, 2:477–521, 1987.

[Farreny and Ghallab, 1987] H. Farreny and M. Ghallab. – *Eléments d'Intelligence Artificielle.* – Hermès, 1987.

[Fiorini and Shiller, 1996] P. Fiorini and Z. Shiller. – Time optimal trajectory planning in dynamic environments. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1553–1558, Minneapolis, MN (US), April 1996.

[Fraichard and Laugier, 1992] Th. Fraichard and C. Laugier. – Kinodynamic planning in a structured and time-varying 2D workspace. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1500–1505, Nice (FR), May 1992.

[Fraichard and Scheuer, 1994] Th. Fraichard and A. Scheuer. – Car-like robots and moving obstacles. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 64–69, San Diego, CA (US), May 1994.

[Fraichard, 1993] Th. Fraichard. – Dynamic trajectory planning with dynamic constraints: a 'state-time space' approach. – In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 1394–1400, Yokohama (JP), July 1993.

[Fujimura and Samet, 1989] K. Fujimura and H. Samet. – A hierarchical strategy for path planning among moving obstacles. – *IEEE Trans. Robotics and Automation*, 5(1):61–69, February 1989.

[Fujimura and Samet, 1990] K. Fujimura and H. Samet. – Motion planning in a dynamic domain. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 324–330, Cincinnatti, OH (USA), May 1990.

[Jacobs *et al.*, 1989] P. Jacobs, G. Heinzinger, J. Canny, and B. Paden. – Planning guaranteed near-time-optimal trajectories for a manipulator in a cluterred workspace. – Research Report ESRC 89-20/RAMP 89-15, Engineering Systems Research Center, Univ. of California., Berkeley, CA (USA), October 1989.

[Kant and Zucker, 1986] K. Kant and S. Zucker. – Toward efficient trajectory planning: the path-velocity decomposition. – *Int. Journal of Robotics Research*, 5(3):72–89, Fall 1986.

[Kyriakopoulos and Saridis, 1991] K. J. Kyriakopoulos and G. N. Saridis. – Collision avoidance of mobile robots in non-stationary environments. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 904–909, Sacramento, CA (US), April 1991.

[Latombe, 1990] J-C. Latombe. – *Robot motion planning.* – Kluwer Academic Press, 1990.

[Lozano-Perez and Wesley, 1979] T. Lozano-Perez and M.A. Wesley. – An algorithm for planning collision-free paths among polyhedral obstacles. – *Communications of the ACM*, 22(10):560–570, October 1979.

[Nilsson, 1980] N.J. Nilsson. – *Principles of artificial intelligence.* – Morgan Kaufmann, Los Altos, CA (USA), 1980.

[Ó'Dúnlaing, 1987] C. Ó'Dúnlaing. – Motion planning with inertial constraints. – *Algorithmica*, 2:431–475, 1987.

[Reif and Sharir, 1985] J. Reif and M. Sharir. – Motion planning in the presence of moving obstacles. – In *Proc. of the IEEE Symp. on the Foundations of Computer Science*, pages 144–154, Portland, OR (USA), October 1985.

[Sahar and Hollerbach, 1985] G. Sahar and J. H. Hollerbach. – Planning of minimum-time trajectories for robot arms. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 751–758, St Louis, MI (USA), March 1985.

[Scheuer and Fraichard, 1997] A. Scheuer and Th. Fraichard. – Continuous-curvature path planning for car-like vehicles. – In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 997–1003, Grenoble (FR), September 1997.

[Shih *et al.*, 1990] C.L. Shih, T.T. Lee, and W.A. Gruver. – Motion planning with time-varying polyhedral obstacles based on graph search and mathematical programming. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 331–337, Cincinnatti, OH (USA), May 1990.

[Shiller and Chen, 1990] Z. Shiller and J.C. Chen. – Optimal motion planning of autonomous vehicles in three-dimensional terrains. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 198–203, Cincinnatti, OH (USA), May 1990.

[Shiller and Dubowsky, 1985] Z. Shiller and S. Dubowsky. – On the optimal control of robotic manipulators with actuator and end-effector constraints. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 614–620, St Louis, MI (USA), March 1985.

[Shiller and Dubowsky, 1988] Z. Shiller and S. Dubowsky. – Global time optimal motions of robotic manipulators in the presence of obstacles. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 370–375, Philadelphia, PA (USA), April 1988.

[Shiller and Dubowsky, 1989] Z. Shiller and S. Dubowsky. – Robot path planning with obstacles, actuator, gripper and payload constraints. – *Int. Journal of Robotics Research*, 8(6):3–18, December 1989.

[Shiller and Lu, 1990] Z. Shiller and H-H. Lu. – Robust computation of path constrained time-optimal motion. – In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 144–149, Cincinnatti, OH (USA), May 1990.

[Shin and McKay, 1985] K. G. Shin and N. D. McKay. – Minimum-time control of robotic manipulators with geometric path constraints. – *IEEE Trans. Autom. Contr.*, 30:531–541, June 1985.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399